REMARKS

Claims 1-39 are rejected under 35 U.S.C. § 101 as allegedly being directed to non-statutory subject matter. Claims 1-10, 13-23, 26-36 and 39 are rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by Lahti (US Patent No. 4,875,161). Upon entry of the present amendment, claims 1-26 and 40-67 will remain in this application. Claims 2, 12, 15, and 25 are amended. Claims 27-39 are cancelled. Claims 40-67 are added.

Formal Matters

Applicants thank the Examiner for pointing out informalities in claims 12 and 25. Both claims have been amended to correct the informalities, in order to overcome the pending objections. In addition, claims 2 and 15 have been amended to make minor typographical corrections.

Rejections Under 35 U.S.C. § 101

Applicants submit that claims 1-26 constitute patentable subject matter under 35 U.S.C. § 101. With regard to claim 1, the Examiner contends that the steps of "decoding" and "providing" amount to an abstract idea. Applicants respectfully disagree. A "process" is one of the statutorily defined categories of inventions that clearly constitute patentable subject matter. 35 U.S.C. § 101 states that any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may be patented. As a method claim, claim 1 falls squarely within the category of "process" inventions. Thus as an initial matter, claim 1 must be recognized as being part of a category of patentable inventions under § 101.

Claim 1 also does not belong to any of the judicially created exceptions to patentable subject matter -- i.e., abstract ideas, laws of nature, and natural phenomenon. Specifically, abstract ideas relate to mathematical expressions and other fundamental principles that are only

expressed in the abstract, such as Einstein's famous equation E = MC2. By contrast, claim 1 relates to a specific method of processing data in a programmable processor. The method comprises the steps of:

> (1) "decoding a single instruction for selectively arranging data, specifying a data selection operand and a first and second registers...the data selection operand comprising a plurality of fields each selecting one of the plurality of data elements"; and
>
> (2) "for each field of the data selection operand, providing the data element selected by the field to a predetermined position in a catenated result."

These steps operate to decode a single instruction and arrange data elements found in a first and second register according to a data selection operand, by providing the data elements selected by each field within the data selection operand to a predetermined position in a catenated result. The steps require specific, physical acts to be performed on the data elements found in registers within the programmable processor. This is not just an abstract idea.

To be sure, claim 1 is not directed to general concepts of "decoding" and "providing," as the Examiner seems to suggest. Rather, claim 1 is directed to the specific steps for decoding a single instruction specifying a data selection operand and first and second registers and providing the data element selected by each field of the data selection operand in a catenated result, as discussed above. That is, the decoding and providing steps of claim 1 are performed in an actual programmable processor, on data elements found in physical registers. Such specific steps go far beyond just amorphous concepts of "decoding" and "providing" envisioned in some abstract form.

Furthermore, the method of claim 1 also has tremendous practical application. Even if an invention touches on an abstract idea, law of nature, and natural phenomenon, the invention may still be patentable if it possesses a practical application. See Diamond v. Diehr, 450 U.S. 175,

187, 209 USPQ 1, 8 (1981) ("[i]t is now commonplace that an application of a law of nature or mathematical formula to a known structure or process may well be deserving of patent protection.") (emphasis in original). The specific decoding and providing steps recited in claim 1 are applied in a practical manner to arrange data elements inside a programmable processor. Thus, even if it is assumed arguendo that the general concepts of decoding and providing somehow touch on abstract ideas, which they do not, the practical application of decoding a single instruction for selectively arranging data elements found in registers and providing selected data elements in a catenated result, all performed in a programmable processor, undoubtedly establishes that the invention of claim 1 would nevertheless possess patentable subject matter.

Indeed, the invention sets forth a powerful technique for arranging data in a programmable processor using a single instruction, Using this technique, software programmers can create new, more versatile computer programs that may not have been feasible before. Since computer programs literally operate everyday to support countless industries, an improvement that allows better computer programs to be written clearly provides a practical application that has useful, tangible, and concrete results. For all of the reasons stated above, the invention recited in claim 1 deserves patent protection. Claims 2-13 depend from claim 1 and therefore also possess patentable subject matter, for at least the same reasons stated above with respect to claim 1.

With regard to claims 14, the Examiner repeats the same objection to claim 1 – i.e., that the features of decoding and providing allegedly represent only an "abstract idea." For at least similar reasons as stated above with respect to claim 1, Applicants submit that the invention of claim 14 is also not an "abstract idea." The invention is not directed to the general concept of

decoding and providing. Rather, the invention relates to a specific technique performed within a programmable processor for arranging data elements in registers by using a single instruction. Furthermore, the invention provides practical application that has useful, tangible, and concrete results.

In addition, the Examiner contends that the "computer-readable medium" recited in claim 14 is not limited to statutory subject matter. Applicants respectfully disagree. The "computer-readable medium" of claim 14 is either an article of manufacture or a machine (or both), which are statutorily recognized categories of patentable inventions under 35 U.S.C. § 101. It is well known in the art that a computer-readable medium may be manufactured to include instructions that instruct a computer to perform various operations. An example is a CD-ROM with installed software instructions. Such a computer-readable medium is an article of manufacture, which is one of the statutorily recognized categories of patentable inventions. In addition, a computer-readable medium such as that recited in claim 14 may also be a machine. For example, the CD-ROM with installed software instructions is also a machine having multiple layers of components that operate together to store digital information. Such a machine is also one of the statutorily recognized categories of patentable inventions. For at least these reasons, Applicants submit that claim 14 is directed to an invention having patentable subject matter. Claims 15-26 depend from claim 14 and therefore also possess patentable subject matter, for at least the same reasons stated above with respect to claim 14.

Rejections Under 35 U.S.C. § 102(b)

Claim 1

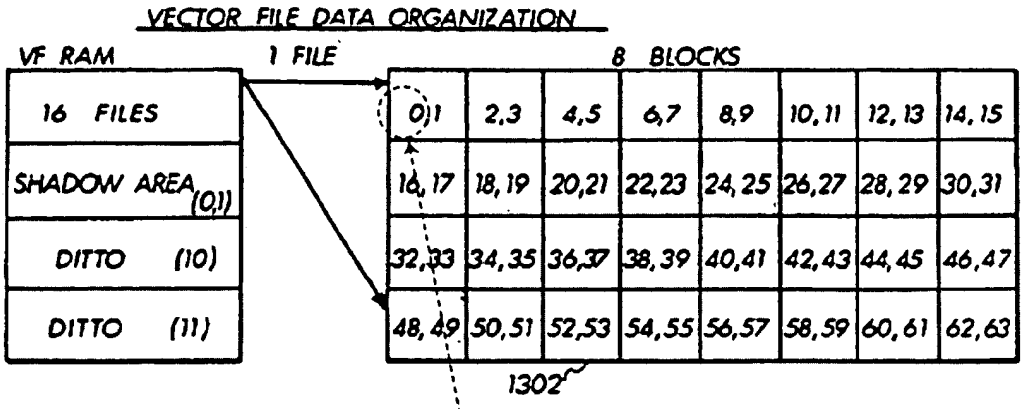
Applicants respectfully assert that Lahti fails to anticipate the invention of claim 1, which recites:

> "decoding a single instruction for selectively arranging data, specifying a data selection operand and a first and a second register each having a register width, the first and second registers providing a plurality of data elements each having an elemental width smaller than the register width, ***the data selection operand comprising a plurality of fields each selecting one of the plurality of data elements***; and
>
> for each field of the data selection operand, providing the data element selected by the field to a predetermined position in a catenated result."

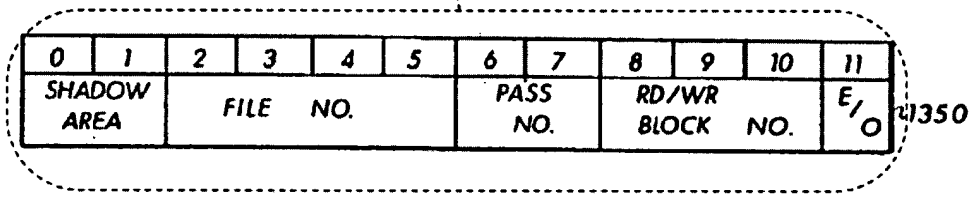
These claim limitations require decoding a single instruction associated with a data selection operand that comprises a plurality of fields each selecting one of the plurality of data elements found in a first and a second register. Lahti fails to disclose such a data selection operand. Instead, Lahti discloses a conventional memory address format comprising a plurality of fields that must be read together to select one word of data. A memory address adhering to Lahti's address format must contain all of its fields, which must be combined together to form a valid memory address that can be used to access a word of data in memory. That is, each field is merely a fractional portion of the complete memory address. Each individual field does not constitute a valid memory address and cannot be used to select any data in Lahti's memory space. Thus, Lahti fails to disclose a data selection operand comprising a plurality of fields, with each field selecting one of a plurality data elements, as required by claim 1.

Lahti's memory space and memory address format are described in more detail below to illustrate this important point. Lahti's memory space is referred to as "vector file data" (shown in Fig. 12 of Lahti), which is accessed using a 12-bit memory address format referred to as the "vector file address format" (shown in Fig. 13 of Lahti). Relevant portions of Figs. 12 and 13 are reproduced below with annotations:

Lahti's Memory Space (Fig. 12)



Lahti's Memory Address Format (Fig. 13)



As shown here, Lahti's memory space is organized into quarters, files, passes, blocks and even/odd words. First, the memory space is divided into 4 quarters. The first quarter (labeled as "16 FILES") is a main memory area, and each of the three remaining quarters (labeled as "SHADOW AREA," "DITTO," and "DITTO") is a shadow memory area used to shadow the contents of the main memory area. See Lahti, col. 18, lines 8-29. The structure of the main memory area is representative of the structure of each of the three shadow memory areas. Next, the main memory area (as well as each of the three shadow memory areas) is divided into 16 "files." In each "file," data is organized into 4 "passes" (rows) and 8 "blocks" (columns). For example, Fig. 12 shows a file 1302 having 4 "passes" (rows) and 8 "blocks" (columns). Finally, for a given "pass" and "block," there is sufficient space to store 2 words of data – an "even"

word and an "odd" word. In this figure, the even words are labeled as "0," "2," "4,"..."62." The odd words are labeled as "1," "3," "5,"..."63."

Thus, each word of data in Lahti's memory space is located in a particular quarter/shadow area, in a particular file within the quarter/shadow area, at a particular pass and block within the file, and at either the even or odd word location found at that particular pass and block. Correspondingly, a valid 12-bit memory address used to access Lahti's memory space includes the following five fields: (1) 2-bit quarter/shadow area, (2) 4-bit file number, (3) 2-bit pass number, (4) 3-bit block number and (5) 1-bit even/odd word bit, as shown in Fig. 13 reproduced above.

All five fields in Lahti's 12-bit memory address format must be taken together to form a valid memory address, which identifies a word of data in the memory space. This is pictorially illustrated in the figures above. Specifically, a dotted arrow (" ") is used to highlight the fact that the entire 12-bit memory address is used to identify a single word of data (in this case, word "0") in the memory space. Any individual field (e.g., the 4-bit "file number" field) taken by itself does not constitute a valid memory address. Indeed, each field is only a fractional portion of the memory address. Therefore, each individual field cannot be used to select a word of data from Lahti's memory space.

Yet that is exactly what the Examiner proposes. The Examiner points to the five fields in Lahti's memory address format as supposedly being capable of "each selecting one of the plurality of data elements," as recited in claim 1. See Office Action dated 4/19/06, p. 5, lines 8-12. Following the Examiner's rationale, each of the five fields in Lahti's memory address format would select a word of data from the memory space. That is, the 2-bit "quarter/shadow area" would select a word of data; the 4-bit "file number" would select a word of data; the 2-bit "pass

number" would select a word of data; the 3-bit "block number" would select a word of data; and finally the 1-bit "even/odd" word bit would select a word of data. Clearly, this is not how Lahti's memory address format operates. The five fields in Lahti's memory address format simply cannot each select a data element.

Indeed, the basic design of the Lahti system actually teaches away from a data selection operand having fields that each selects a data element. The Lahti system is designed for efficient vector read/write operations that retain the order of the data elements as they are stored in the memory space. See Lahti, col. 19, lines 7-63 (describing Lahti's vector read/write operation as it is performed sequentially in the order of the data elements, on successively clock cycles). These vector file read/write operations always start at an address zero of a file of data elements and proceeds sequentially through the data elements, in the same order as the data is stored. See Lahti, col. 17, lines 54-56 ("Vector File reads or writes always start at address zero of the file and continue sequential addresses") (emphasis added). Because Lahti's read/write operations always start at an initial data element and always occur in order, only one data element needs to be selected to accomplish the operation – the first data element. Selecting any more than the first data element is actually unnecessary and wasteful. Thus, the basic design of the Lahti system in fact teaches away from the use of a data selection operand with multiple fields that each selects a data element, as recited in claim 1. Given the various reasons stated above, Applicants respectfully assert that Lahti fails to teach or suggest the invention as recited in claim 1.

Claim 2

Claim 2 depends from claim 1 and therefore incorporates all of its limitations. As such, for at least the reasons discussed above with respect to claim 1, Lahti also fails to anticipate claim 2. Furthermore, claim 2 recites:

> "wherein each field of the data selection operand provides a sufficient number of bits to specify any one of the plurality of data elements"

Lahti also fails to disclose this claim limitation because each field in Lahti's memory address format actually provides an insufficient number of bits to specify data within Lahti's memory space. As discussed previously, all 12 bits of the memory address format are required to select any word of data from Lahti's memory space. Each of the five fields in this 12-bit memory address format contains less than 12 bits. Clearly, each individual field has an insufficient number of bits to select any word of data from the memory space.[1] For at least these reasons, Lahti fails to anticipate claim 2.

Claim 3

Claim 3 depends from claim 1 and therefore incorporates all of its limitations. As such, for at least the reasons discussed above with respect to claim 1, Lahti also fails to anticipate claim 3. Furthermore, claim 3 recites:

> "wherein each field of the data selection operand has a width of n bits, wherein the plurality of data elements comprises 2n data elements"

Lahti fails to disclose a data selection operand containing fields that each has a width of n bits, wherein the plurality of data elements comprises 2n data elements. The Examiner merely cites to a passage in Lahti as stating that 26 = 64 words are selected. See Office Action dated 4/19/06, p. 6, second paragraph. Following the Examiner's reasoning, the value "n" would equal

[1] Even when the words are isolated within a pair of blocks, it is still impossible for any given field within the 12-bit memory address format to provide a sufficient number of bits to select any one of the words. The Examiner points to a pair of blocks, labeled as "BLOCK 0" and "BLOCK 1" in Fig. 13, as supposedly containing the "plurality of data elements" recited in claim 2. See Office Action dated 4/19/06, p. 5, lines 3-6 (discussing the "plurality of data elements" with respect to claim 1, the limitations of which are incorporated in claim 2). Here, BLOCK 0 contains words 0, 1, 16, 17, 32, 33, 48 and 49. BLOCK 1 contains words 2, 3, 18, 19, 34, 35, 50 and 51. See Lahti, Fig. 13. At least three fields must be specified in order to address any one of these words in BLOCK 0 and BLOCK 1. Specifically, the 3-bit "block number" field must be specified to select either BLOCK 0 or BLOCK 1. Also, the 2-bit "pass number" field must be specified to select the pass (row) of the word being selected. Furthermore, the 1-bit "even/odd" field must be specified to select the even or odd word. All of these fields are needed. Any one field taken alone does not have enough bits to specify any word within these two blocks. Thus, it remains the case that each field in Lahti's memory address format has an insufficient number of bits to select any one of the words in the memory space, even when the words are isolated to a pair of blocks within the memory space.

to 6 bits. However, none of the five fields in Lahti's memory address format is 6 bits wide (The widest of the five fields is the "file number," which is only 4 bits wide). Thus, the example of "n" being defined as 6 bits (26 = 64) actually contradicts the claim requirement that "each field of the data selection operand has a width of n bits." For at least this additional reason and the reasons stated above, Lahti does not anticipate claim 3.

Claim 4

Claim 4 depends from claim 1 and therefore incorporates all of its limitations. As such, for at least the reasons discussed above with respect to claim 1, Lahti also fails to anticipate claim 4.

Claim 5

Claim 5 depends from claim 1 and therefore incorporates all of its limitations. As such, for at least the reasons discussed above with respect to claim 1, Lahti also fails to anticipate claim 5. Furthermore, claim 5 recites:

> "wherein the data selection operand has a width equal to the specified register width"

Lahti fails to disclose a data selection operand having a width equal to the specified register width. Here, the "specified register width" refers to the width of a register specified by a single instruction.[2] Lahti's 12-bit memory address format is the only feature the Examiner points to as supposedly being a "data selection operand." See Office Action dated 4/13/06, p. 5, lines 2-3. As shown in Fig. 13, Lahti's memory address format is exactly 12 bits wide. However, Lahti simply does not disclose any register, specified by a single instruction, that has the same width – 12 bits wide. The Examiner points to the widths of words 0 and 1 in BLOCK 0

[2] Claim 4 , from which claim 5 depends, recites "wherein the data selection register is provided by a register specified by the single instruction" (emphasis added). Thus, the "specified register width" refers to the width of the register specified by the single instruction.

shown in Fig. 13 of Lahti. See Office Action dated 4/19/06, p. 6 last paragraph and p. 7, first paragraph. However, the individual width of word 0, as well as word 1, is 36 bits – not 12 bits. See Lahti, col. 3, lines 38 ("Each file has space for sixty-four 36-bit words). In other words, the alleged "data selection operand" has a width (12 bits) that is NOT equal to the width of the alleged "specified register" (36 bits). Thus, even under the Examiner's own reasoning, Lahti not only fails to disclose but in fact explicitly teaches away from the feature recited in claim 5. For this additional reason and the reasons stated above, Lahti cannot anticipate claim 5.

Claim 6

Claim 6 depends from claim 1 and therefore incorporates all of its limitations. As such, for at least the reasons discussed above with respect to claim 1, Lahti also fails to anticipate claim 6.

Claim 7

Claim 7 depends from claim 1 and therefore incorporates all of its limitations. As such, for at least the reasons discussed above with respect to claim 1, Lahti also fails to anticipate claim 7.

Claim 8

Claim 8 depends from claim 1 and therefore incorporates all of its limitations. As such, for at least the reasons discussed above with respect to claim 1, Lahti also fails to anticipate claim 8. Furthermore, claim 8 recites:

> "wherein the instruction further specifies a data element width of the plurality of data elements"

Lahti fails to teach this claimed feature. There is absolutely no disclosure of any instruction that specifies a data element width. The Examiner merely points to "words 0-63" shown in Fig. 13 or Lahti. See Office Action dated 4/19/06, p. 7, paragraph 4. However, the

Examiner says nothing about the how the width of "words 0-63" is specified – i.e., whether the width is specified by an instruction. Indeed, there is no teaching in Lahti regarding any instruction that specifies the width of words such as words 0-63 in the memory space. For at least this additional reason and the reasons stated previously, Lahti cannot anticipate claim 8.

Claim 9

Claim 9 depends from claim 1 and therefore incorporates all of its limitations. As such, for at least the reasons discussed above with respect to claim 1, Lahti also fails to anticipate claim 9. Furthermore, claim 9 recites:

"wherein each data element has a width of 8 bits"

Lahti fails to teach this claim limitation. The Examiner insists that in Lahti, "each word can be 8 bits in length." See Office Action dated 4/19/06, p. 7, last paragraph. However, the Examiner provides absolutely no support for this statement. Lahti simply contains no teaching or suggestion of a word width of 8 bits. In fact, the width of each word in Lahti's memory space is 36 bits, not 8 bits. See Lahti, col. 3, lines 38 ("Each file has space for sixty-four 36-bit words"). As such, Lahti expressly teaches away from a data element width of 8 bits. For at least this additional reason, Lahti cannot anticipate claim 9.

Claim 10

Claim 10 depends from claim 1 and therefore incorporates all of its limitations. As such, for at least the reasons discussed above with respect to claim 1, Lahti also fails to anticipate claim 10.

Claims 13, 14 and 26

Each of claims 13, 14 and 26 is rejected based on the same rationale as claim 1. For at least the reasons discussed above with respect to claim 1, Lahti also fails to anticipate claims 13, 14 and 26.

Claims 15-23

Claims 15-23 are rejected based on the same rationale as claims 2-10. For at least the reasons discussed above with respect to claims 2-10, Lahti also fails to anticipate claims 15-23.

Support for Claims 40-67

New claims 40-67 are fully supported by the present specification. Support for each claim is identified below by citing to the present specification as published (United States Pub. No. US2004/0153632).

Regarding claim 40, a method of processing data in a programmable processor comprising "decoding a single instruction specifying a plurality of registers storing a plurality of 8-bit data elements, an index register storing an index vector comprising a plurality of equal-sized selectors stored in partitioned fields of the index register and a destination register; and for each selector in the index vector, providing a data element selected by the selector to a predetermined position in the destination register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 41, the recited claim feature of "wherein the plurality of registers comprises two registers" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 42, the recited claim feature of "wherein the plurality of registers comprises two 64-bit registers storing a combined total of sixteen 8-bit data elements" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 43, the recited claim feature of "wherein the number of selectors stored in the index register is equal to the number of predetermined positions in the destination register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 44, the recited claim feature of "wherein the index register is a 64-bit register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 45, the recited claim feature of "wherein the index vector comprises n equal-sized selectors and the destination register comprises n equal-sized predetermined positions" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 46, the recited claim feature of "wherein the selector stored in a lowest order set of bits of the index register provides a data element to a lowest order set of bits of the destination register, the selector in a second lowest order set of bits of the index register provide a data element to a second lowest order set of bits of the destination register and the selector stored in a highest order set of bits of the index register provides a data element to a highest order set of bits of the destination register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 47, the recited claim feature of "wherein the destination register is a 128-bit register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 48, the recited claim feature of "wherein each of the equal-sized selectors stored in partitioned fields of the index register is a 4-bit selector" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 49, the recited claim feature of "wherein the index register stores sixteen 4-bit selectors" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 50, a method of processing data in a programmable processor comprising "decoding a single instruction specifying a first register storing a first plurality of 8-bit data elements, a second register storing a second plurality of 8-bit data elements, an index register storing an index vector comprising a plurality of equal-sized selectors stored in

partitioned fields of the index register and a destination register; for each selector in the index vector, providing a data element from one of the first or second plurality of 8-bit data elements selected by the selector to a predetermined 8-bit position in the destination register, wherein the predetermined positions are contiguous blocks of bits that take up an entire width of the destination register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 51, the recited claim feature of "wherein the first and second registers are 64-bit registers, the index register is a 64-bit register and each selector stored in the index register has a sufficient number of bits to select any one of the 8-bit data elements in the first or second pluralities of 8-bit data elements" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 52, the recited claim feature of "wherein the destination register is a 128-bit register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 53, the recited claim feature of "wherein each of the equal-sized selectors stored in partitioned fields of the index register is a 4-bit selector" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 54, a computer-readable medium having stored therein a plurality of instructions comprising in part "an instruction specifying a plurality of registers storing a plurality of 8-bit data elements, an index register storing an index vector comprising a plurality of equal-sized selectors stored in partitioned fields of the index register and a destination register; and wherein for each selector in the index vector, the instruction causes the computer processor to provide a data element selected by the selector to a predetermined position in the destination register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 55, the recited claim feature of "wherein the plurality of registers comprises two registers" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 56, the recited claim feature of "wherein the plurality of registers comprises two 64-bit registers storing a combined total of sixteen 8-bit data elements" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 57, the recited claim feature of "wherein the number of selectors stored in the index register is equal to the number of predetermined positions in the destination register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 58, the recited claim feature of "wherein the index register is a 64-bit register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 59, the recited claim feature of "wherein the index vector comprises n equal-sized selectors and the destination register comprises n equal-sized predetermined positions" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 60, the recited claim feature of "wherein the selector stored in a lowest order set of bits of the index register provides a data element to a lowest order set of bits of the destination register, the selector in a second lowest order set of bits of the index register provide a data element to a second lowest order set of bits of the destination register and the selector stored in a highest order set of bits of the index register provides a data element to a highest order set of bits of the destination register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 61, the recited claim feature of "wherein the destination register is a 128-bit register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 62, the recited claim feature of "wherein each of the equal-sized selectors stored in partitioned fields of the index register is a 4-bit selector" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 63, the recited claim feature of "wherein the index register stores sixteen 4-bit selectors" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 64, computer-readable medium having stored therein a plurality of instructions comprising "an instruction specifying a first register storing a first plurality of 8-bit data elements, a second register storing a second plurality of 8-bit data elements, an index register storing an index vector comprising a plurality of equal-sized selectors stored in partitioned fields of the index register and a destination register; and wherein for each selector in the index vector, the instruction causes the computer processor to provide a data element from one of the first or second plurality of 8-bit data elements selected by the selector to a predetermined 8-bit position in the destination register, wherein the predetermined positions are contiguous blocks of bits that take up an entire width of the destination register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 65, the recited claim feature of "wherein the first and second registers are 64-bit registers, the index register is a 64-bit register and each selector stored in the index register has a sufficient number of bits to select any one of the 8-bit data elements in the first or second pluralities of 8-bit data elements" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 66, the recited claim feature of "wherein the destination register is a 128-bit register" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

Regarding claim 67, the recited claim feature of "wherein each of the equal-sized selectors stored in partitioned fields of the index register is a 4-bit selector" is described at Figures 47D and 47E, and paragraphs 0078 and 0301.

CONCLUSION

Accordingly, it is believed that all pending claims are now in condition for allowance. Applicants therefore respectfully request an early and favorable reconsideration and allowance of this application. If there are any outstanding issues which might be resolved by an interview or an Examiner's amendment, the Examiner is invited to call Applicants' representative at the telephone number shown below.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 500417 and please credit any excess fees to such deposit account.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP

Demetria A. Buncum
Registration No. 58,848

600 13th Street, N.W.
Washington, DC 20005-3096
Phone: 202.756.8000 DAB:jrj
Facsimile: 202.756.8087
Date: January 25, 2007

Please recognize our Customer No. 20277 as our correspondence address.